

LONG ISLAND GROUP

Sinclair

IMEX

8/84

ITEM	PAGE
I. MEETING	1
II. ENLARGED CHARACTER MODES 2068 by Jeff Street	2
III. A WHALE OF AN IDEA - USING TS 2068 IN BUSINESS - by Harvey Rait	4
IV. ZX/PROFILE IMPROVEMENT by Kevin Smith	4
V. Unplotting on the 2068 by Jeff Street	5
VI. CATALOG /SOFTWARE RECEIVED	6

MEETING NOTES

Our last meeting was held on July 7th at Heinz H's place. Agenda was a little loose, but here is what we did:

1. Harvey R. demoed some of the attention getting displays he's developed for use in his business (see Listing, P-4).
2. Kevin Smith showed us his Panasonic "Data" recorder. These sell for \$40. at DOA and provide a counter, automatic tone control ("data" setting) and phase reversal switch. A very well priced machine.
3. Heinz had set up his "shaper" (preliminary schematic published previously) which really shapes up the TS 1000's waveforms. Coming out of the "shaper", they all looked text-book perfect. We'll have more details on this device in a latter issue.
4. Bob M. brought along his Westridge Modem and we called the LICA bulletin board. We left a message for Al Levy to call us and browsed around a bit. This BBS is a big one and seems quite well run. We tried a few other BBS's with mixed-to-poor results. The Modem hardware seems excellent (auto dial is a nice feature), but good software just isn't here yet.
5. I tried "Firstloader" several times and we got a full run through, but apparently picked up some garbage along the way. At any rate, a working copy of FirstLoadr was included on the Library Tape #1.
6. Library tape #1 was distributed. It includes several public domain games (from U.K. magazines), FIRSTLOADR, my memory list program (used to crack Transylvania Tower), Jeff S's big character and uplot routines and some graphics demo programs. Bring a blank tape (and your recorder) to the next meeting if you want a copy.

Nazir P. was not at the meeting, but he has produced a simple word processor and cracked the other Spectrum program, "Fantasia" and will demo these, next meeting. Speaking of which:

→ Next meeting is scheduled for August 26th (Sunday) at 3:00PM. We decided at the last meeting that we need to have a formal "business" portion in our meetings from now on, and an agenda for the "show and tell".

The proposed agenda for next meeting then is:

- 1) Business Meeting
 - a) Call to Order and Review of last meeting
 - b) Status of the Treasury
 - c) Review of correspondence generated by and received by the group
 - d) Plan for next meeting (timing, agenda, location)
 - e) Motions from the floor (suggestions to buy hardware/software, funding for meeting site hosts, establishment of committees, election of officers, production of a charter, etc.)
- 2) Demonstration Phase
 - a) "Fantasia" game and word Processor - Nazir P.
 - b) Quickload/Fastload/Q Save/XLR8 - which is best?
We will have power and a few TV's and TS 1000's set up.
Bring your favorite quick-loading program (I have Q-Save, for example) and we'll run Benchmark tests to see which is the most reliable, fastest, etc.
- 3) General Discussion

If you've got a new toy (hardware or software), or typed in a program from a magazine, or have "rolled your own" program for a particular need, please bring it and share it with us.

Enlarged Character Modes - 2068

The ability to print characters larger than normal size, easily and rapidly, could substantially enhance the text and graphical displays of many programs. I created the following machine code routine, and associated procedures, with this ability in mind. The routine allows printing of three enlarged character modes from BASIC...

Regular Characters

Mode 2 Characters

Mode 1 Char\$

Mode 3 Char\$

MODE	HEIGHT	x	WIDTH	MAXIMUM CHAR/LN
1	1	x	2	16
2	2	x	1	31
3	2	x	2	16

Printing these character modes from BASIC will entail creating and SAVEing a two line BASIC program (which includes machine code in a REM statement). Then, you can either MERGE it with an existing program or create a new program around it. With these two lines resident in a program you 'Print' the enlarged modes using the following procedure...

```
POKE 23728,line
POKE 23729,column
POKE 23723,mode
RANDOMIZE USA FN a()
REM "char string"
```

...Just POKE the values for the mode you desire, and the line and column at which you want printing to start, into the appropriate addresses. Then call the MC routine using the RANDOMIZE USA statement followed directly by a REM statement with

your string enclosed in quotes. You can think of this procedure as a hypothetical BASIC statement...

```
PRINT MODE n; AT ln,cl;"string"
```

Line and column values are the same as normally used but when referring to the enlarged modes you are specifying the start position for the upper, leftmost portion of the character. For instance; if line=2, column=2, mode=3, and string="XTEST"...

```
column
23456789
line 2
line 3XTEST
```

To create the routine, input the loader program below and RUN it...

```
1 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2 DEF FN a()=PEEK 23635+255*P
EEK 23635+5
10 LET x=0: LET sum=0
14 ON ERR GO TO 70
20 INPUT a$
22 LET dec=VAL a$
30 POKE FN a() + x,dec
35 PRINT a$;" "
40 LET x=x+1
50 LET sum=sum+dec
60 GO TO 20
70 ON ERR RESET
75 PRINT
80 PRINT "sum=";sum
```

Enlarged Character Modes - 2068

...it is waiting for you to input the decimal code listing; do so very carefully, you would not want to do it twice.

```

033:175:092:126:033:175:092:254:
007:040:008:254:015:040:004:203:
134:024:002:203:198:042:085:092:
001:004:000:009:062:034:190:033:
040:002:024:250:229:033:177:092:
078:006:000:033:000:054:009:229:
033:175:092:126:200:007:111:062:
006:038:000:006:005:003:037:203:
023:016:250:103:193:000:229:033:
176:092:001:000:008:126:225:254:
008:056:006:254:016:056:001:009:
009:034:173:092:225:229:126:001:
000:061:197:254:144:055:008:006:
112:144:193:001:088:255:197:006:
032:144:111:062:000:005:003:203:
037:203:023:016:250:103:193:009:
229:033:171:092:126:225:254:001:
040:114:024:002:024:207:062:008:
050:172:092:229:126:087:030:000:
033:171:092:126:254:003:122:040:
123:006:002:042:173:092:197:001:
000:001:000:193:114:035:115:043:
050:172:092:254:001:040:037:254:
005:032:031:190:254:002:032:006:
229:033:175:092:203:070:225:197:
032:009:001:224:007:167:237:086:
193:024:007:001:224:000:167:237:
068:193:016:092:034:173:092:225:
035:058:172:092:061:050:172:092:
032:169:033:173:092:203:070:042:
173:092:040:012:001:000:007:167:
237:066:024:004:024:065:024:140:
001:031:007:167:237:066:017:171:
092:026:254:003:032:001:035:034:
173:092:225:035:062:034:190:200:
024:228:024:133:006:004:033:000:
000:203:063:003:029:203:045:016:
248:006:004:203:063:203:028:203:
044:016:248:004:093:033:171:092:
126:254:001:040:009:024:219:006:
008:197:229:126:024:214:042:173:
092:114:035:115:043:001:000:001:
009:034:173:092:225:035:193:016:
232:042:173:092:001:254:007:167:
237:066:034:173:092:225:035:062:
034:190:200:024:145:120:120:120:
120:120:120:

```

Sum=34865

When you've entered the last decimal code, input an "s"; the program will respond by printing a value called 'Sum'. It should equal 34865. If it doesn't you will have to reinput the code. If you got the correct value, delete program lines 10 through 80 and SAVE the remaining lines. With line 1 and 2 still resident in the computer you are ready to experiment with the procedure I outlined earlier for generating the enlarged characters. But before you do...

You will notice that a manual or automatic attempt at LISTing the line 1 REM statement causes a spurious error report. This occurs because the MC code contains values that the system uses for color code markers. This causes annoying interference with program development and editing. Luckily we can circumvent the problem by doing the following...

1. LIST 2
2. POKE 23660,30
3. Avoid the following...
 - a. LIST
 - b. LIST 1
 - c. Moving the edit cursor onto line 1

Also, a few words of caution; I included no error trapping in the mc routine. Line and column values out of range could be disastrous, mode values other than 1-3 default printing to mode 2. Wrap around printing was not explicitly designed into the code. Mode 1 will usefully exhibit this behavior, the others will not.

WHERE TO BUY IT

Here is 47th Street Photo's latest ad:

TIMEX

TIMEX 2068 PACKAGE
 72K Color Computer • 2020 Recorder •
 Alphacom 3 40-column Printer • Two
 extra software cassettes.
 Orig. \$350
 Now Only **\$179⁹⁵**

TIMEX TS-1500
 16K Computer
 Now Only **\$59⁹⁵**

ALPHACOM 40 column printer
 for all Timex computers **\$49.95**

LONG
ISLAND GROUP
Sinclair
TIMEX

A WHALE OF AN IDEA

USERS GOING TS 2068 IN BUSINESS

HARVEY RAIT

Harvey Rait runs his own small business, a Carvel Ice Cream store, in Valley Stream. To attract customers attention, and at the same time entertain and inform them about his latest promotions, sales, etc., Harvey uses the outstanding color graphics capabilities of the Times 2068.

An example of one of many special programs which Harvey has written is shown below. Note how he has made very good use of the Draw, Plot and Circle commands.

Just about any retail business should be able to benefit from using the TS 2068's "advertizing" capabilities.

```
22 INK 2
24 PRINT "          WHALE-OF-A-CAKE
25 PLOT 240,70
26 DRAW -40,64,.5*PI
27 DRAW -10,-30
28 DRAW -90,40,.5*PI
29 DRAW 0,-50,PI
30 DRAW 0,-20,PI
31 DRAW 90,20,.5*PI
32 DRAW 50,-24
33 CIRCLE 100,120,5
34 INK 2: CIRCLE 100,120,3
50 FOR I=-20 TO 20 STEP .5
51 PLOT 1*6+127,70+5*5IN I
52 NEXT I
55 PRINT INK 2 AT 13,1:" FUGGIE THE
60 PRINT INK 2," FUGGIE THE
```

```
PRICED AT $9.95 IT'S THE LOWEST
PRICE AROUND
WHALE-OF-A-CAKE
AT $9.95 IT'S THE LOWEST PRICE
AROUND
70 COPY
71 CLS
72 GO TO 60
```



ZX PRO/FILE FIX - KEVIN SMITH

There are times when even ZX PRO/FILE misses a sort, particularly when you're looking through long strings of numbers, reports Kevin Smith. PRO/FILE only compares 2 digits at a time.

Kevin's Fix for that little problem is the following piece of machine code. You can add the program lines shown, or just POKE the values in from immediate mode.

```
400000 RAM
400001 RAM
400002 RAM
400003 RAM
400004 RAM
400005 RAM
400006 RAM
400007 RAM
400008 RAM
400009 RAM
400010 RAM
400011 RAM
400012 RAM
400013 RAM
400014 RAM
400015 RAM
400016 RAM
400017 RAM
400018 RAM
400019 RAM
400020 RAM
400021 RAM
400022 RAM
400023 RAM
400024 RAM
400025 RAM
400026 RAM
400027 RAM
400028 RAM
400029 RAM
400030 RAM
400031 RAM
400032 RAM
400033 RAM
400034 RAM
400035 RAM
400036 RAM
400037 RAM
400038 RAM
400039 RAM
400040 RAM
400041 RAM
400042 RAM
400043 RAM
400044 RAM
400045 RAM
400046 RAM
400047 RAM
400048 RAM
400049 RAM
400050 RAM
400051 RAM
400052 RAM
400053 RAM
400054 RAM
400055 RAM
400056 RAM
400057 RAM
400058 RAM
400059 RAM
400060 RAM
400061 RAM
400062 RAM
400063 RAM
400064 RAM
400065 RAM
400066 RAM
400067 RAM
400068 RAM
400069 RAM
400070 RAM
400071 RAM
400072 RAM
400073 RAM
400074 RAM
400075 RAM
400076 RAM
400077 RAM
400078 RAM
400079 RAM
400080 RAM
400081 RAM
400082 RAM
400083 RAM
400084 RAM
400085 RAM
400086 RAM
400087 RAM
400088 RAM
400089 RAM
400090 RAM
400091 RAM
400092 RAM
400093 RAM
400094 RAM
400095 RAM
400096 RAM
400097 RAM
400098 RAM
400099 RAM
400100 RAM
400101 RAM
400102 RAM
400103 RAM
400104 RAM
400105 RAM
400106 RAM
400107 RAM
400108 RAM
400109 RAM
400110 RAM
400111 RAM
400112 RAM
400113 RAM
400114 RAM
400115 RAM
400116 RAM
400117 RAM
400118 RAM
400119 RAM
400120 RAM
400121 RAM
400122 RAM
400123 RAM
400124 RAM
400125 RAM
400126 RAM
400127 RAM
400128 RAM
400129 RAM
400130 RAM
400131 RAM
400132 RAM
400133 RAM
400134 RAM
400135 RAM
400136 RAM
400137 RAM
400138 RAM
400139 RAM
400140 RAM
400141 RAM
400142 RAM
400143 RAM
400144 RAM
400145 RAM
400146 RAM
400147 RAM
400148 RAM
400149 RAM
400150 RAM
400151 RAM
400152 RAM
400153 RAM
400154 RAM
400155 RAM
400156 RAM
400157 RAM
400158 RAM
400159 RAM
400160 RAM
400161 RAM
400162 RAM
400163 RAM
400164 RAM
400165 RAM
400166 RAM
400167 RAM
400168 RAM
400169 RAM
400170 RAM
400171 RAM
400172 RAM
400173 RAM
400174 RAM
400175 RAM
400176 RAM
400177 RAM
400178 RAM
400179 RAM
400180 RAM
400181 RAM
400182 RAM
400183 RAM
400184 RAM
400185 RAM
400186 RAM
400187 RAM
400188 RAM
400189 RAM
400190 RAM
400191 RAM
400192 RAM
400193 RAM
400194 RAM
400195 RAM
400196 RAM
400197 RAM
400198 RAM
400199 RAM
400200 RAM
400201 RAM
400202 RAM
400203 RAM
400204 RAM
400205 RAM
400206 RAM
400207 RAM
400208 RAM
400209 RAM
400210 RAM
400211 RAM
400212 RAM
400213 RAM
400214 RAM
400215 RAM
400216 RAM
400217 RAM
400218 RAM
400219 RAM
400220 RAM
400221 RAM
400222 RAM
400223 RAM
400224 RAM
400225 RAM
400226 RAM
400227 RAM
400228 RAM
400229 RAM
400230 RAM
400231 RAM
400232 RAM
400233 RAM
400234 RAM
400235 RAM
400236 RAM
400237 RAM
400238 RAM
400239 RAM
400240 RAM
400241 RAM
400242 RAM
400243 RAM
400244 RAM
400245 RAM
400246 RAM
400247 RAM
400248 RAM
400249 RAM
400250 RAM
400251 RAM
400252 RAM
400253 RAM
400254 RAM
400255 RAM
400256 RAM
400257 RAM
400258 RAM
400259 RAM
400260 RAM
400261 RAM
400262 RAM
400263 RAM
400264 RAM
400265 RAM
400266 RAM
400267 RAM
400268 RAM
400269 RAM
400270 RAM
400271 RAM
400272 RAM
400273 RAM
400274 RAM
400275 RAM
400276 RAM
400277 RAM
400278 RAM
400279 RAM
400280 RAM
400281 RAM
400282 RAM
400283 RAM
400284 RAM
400285 RAM
400286 RAM
400287 RAM
400288 RAM
400289 RAM
400290 RAM
400291 RAM
400292 RAM
400293 RAM
400294 RAM
400295 RAM
400296 RAM
400297 RAM
400298 RAM
400299 RAM
400300 RAM
400301 RAM
400302 RAM
400303 RAM
400304 RAM
400305 RAM
400306 RAM
400307 RAM
400308 RAM
400309 RAM
400310 RAM
400311 RAM
400312 RAM
400313 RAM
400314 RAM
400315 RAM
400316 RAM
400317 RAM
400318 RAM
400319 RAM
400320 RAM
400321 RAM
400322 RAM
400323 RAM
400324 RAM
400325 RAM
400326 RAM
400327 RAM
400328 RAM
400329 RAM
400330 RAM
400331 RAM
400332 RAM
400333 RAM
400334 RAM
400335 RAM
400336 RAM
400337 RAM
400338 RAM
400339 RAM
400340 RAM
400341 RAM
400342 RAM
400343 RAM
400344 RAM
400345 RAM
400346 RAM
400347 RAM
400348 RAM
400349 RAM
400350 RAM
400351 RAM
400352 RAM
400353 RAM
400354 RAM
400355 RAM
400356 RAM
400357 RAM
400358 RAM
400359 RAM
400360 RAM
400361 RAM
400362 RAM
400363 RAM
400364 RAM
400365 RAM
400366 RAM
400367 RAM
400368 RAM
400369 RAM
400370 RAM
400371 RAM
400372 RAM
400373 RAM
400374 RAM
400375 RAM
400376 RAM
400377 RAM
400378 RAM
400379 RAM
400380 RAM
400381 RAM
400382 RAM
400383 RAM
400384 RAM
400385 RAM
400386 RAM
400387 RAM
400388 RAM
400389 RAM
400390 RAM
400391 RAM
400392 RAM
400393 RAM
400394 RAM
400395 RAM
400396 RAM
400397 RAM
400398 RAM
400399 RAM
400400 RAM
400401 RAM
400402 RAM
400403 RAM
400404 RAM
400405 RAM
400406 RAM
400407 RAM
400408 RAM
400409 RAM
400410 RAM
400411 RAM
400412 RAM
400413 RAM
400414 RAM
400415 RAM
400416 RAM
400417 RAM
400418 RAM
400419 RAM
400420 RAM
400421 RAM
400422 RAM
400423 RAM
400424 RAM
400425 RAM
400426 RAM
400427 RAM
400428 RAM
400429 RAM
400430 RAM
400431 RAM
400432 RAM
400433 RAM
400434 RAM
400435 RAM
400436 RAM
400437 RAM
400438 RAM
400439 RAM
400440 RAM
400441 RAM
400442 RAM
400443 RAM
400444 RAM
400445 RAM
400446 RAM
400447 RAM
400448 RAM
400449 RAM
400450 RAM
400451 RAM
400452 RAM
400453 RAM
400454 RAM
400455 RAM
400456 RAM
400457 RAM
400458 RAM
400459 RAM
400460 RAM
400461 RAM
400462 RAM
400463 RAM
400464 RAM
400465 RAM
400466 RAM
400467 RAM
400468 RAM
400469 RAM
400470 RAM
400471 RAM
400472 RAM
400473 RAM
400474 RAM
400475 RAM
400476 RAM
400477 RAM
400478 RAM
400479 RAM
400480 RAM
400481 RAM
400482 RAM
400483 RAM
400484 RAM
400485 RAM
400486 RAM
400487 RAM
400488 RAM
400489 RAM
400490 RAM
400491 RAM
400492 RAM
400493 RAM
400494 RAM
400495 RAM
400496 RAM
400497 RAM
400498 RAM
400499 RAM
400500 RAM
400501 RAM
400502 RAM
400503 RAM
400504 RAM
400505 RAM
400506 RAM
400507 RAM
400508 RAM
400509 RAM
400510 RAM
400511 RAM
400512 RAM
400513 RAM
400514 RAM
400515 RAM
400516 RAM
400517 RAM
400518 RAM
400519 RAM
400520 RAM
400521 RAM
400522 RAM
400523 RAM
400524 RAM
400525 RAM
400526 RAM
400527 RAM
400528 RAM
400529 RAM
400530 RAM
400531 RAM
400532 RAM
400533 RAM
400534 RAM
400535 RAM
400536 RAM
400537 RAM
400538 RAM
400539 RAM
400540 RAM
400541 RAM
400542 RAM
400543 RAM
400544 RAM
400545 RAM
400546 RAM
400547 RAM
400548 RAM
400549 RAM
400550 RAM
400551 RAM
400552 RAM
400553 RAM
400554 RAM
400555 RAM
400556 RAM
400557 RAM
400558 RAM
400559 RAM
400560 RAM
400561 RAM
400562 RAM
400563 RAM
400564 RAM
400565 RAM
400566 RAM
400567 RAM
400568 RAM
400569 RAM
400570 RAM
400571 RAM
400572 RAM
400573 RAM
400574 RAM
400575 RAM
400576 RAM
400577 RAM
400578 RAM
400579 RAM
400580 RAM
400581 RAM
400582 RAM
400583 RAM
400584 RAM
400585 RAM
400586 RAM
400587 RAM
400588 RAM
400589 RAM
400590 RAM
400591 RAM
400592 RAM
400593 RAM
400594 RAM
400595 RAM
400596 RAM
400597 RAM
400598 RAM
400599 RAM
400600 RAM
400601 RAM
400602 RAM
400603 RAM
400604 RAM
400605 RAM
400606 RAM
400607 RAM
400608 RAM
400609 RAM
400610 RAM
400611 RAM
400612 RAM
400613 RAM
400614 RAM
400615 RAM
400616 RAM
400617 RAM
400618 RAM
400619 RAM
400620 RAM
400621 RAM
400622 RAM
400623 RAM
400624 RAM
400625 RAM
400626 RAM
400627 RAM
400628 RAM
400629 RAM
400630 RAM
400631 RAM
400632 RAM
400633 RAM
400634 RAM
400635 RAM
400636 RAM
400637 RAM
400638 RAM
400639 RAM
400640 RAM
400641 RAM
400642 RAM
400643 RAM
400644 RAM
400645 RAM
400646 RAM
400647 RAM
400648 RAM
400649 RAM
400650 RAM
400651 RAM
400652 RAM
400653 RAM
400654 RAM
400655 RAM
400656 RAM
400657 RAM
400658 RAM
400659 RAM
400660 RAM
400661 RAM
400662 RAM
400663 RAM
400664 RAM
400665 RAM
400666 RAM
400667 RAM
400668 RAM
400669 RAM
400670 RAM
400671 RAM
400672 RAM
400673 RAM
400674 RAM
400675 RAM
400676 RAM
400677 RAM
400678 RAM
400679 RAM
400680 RAM
400681 RAM
400682 RAM
400683 RAM
400684 RAM
400685 RAM
400686 RAM
400687 RAM
400688 RAM
400689 RAM
400690 RAM
400691 RAM
400692 RAM
400693 RAM
400694 RAM
400695 RAM
400696 RAM
400697 RAM
400698 RAM
400699 RAM
400700 RAM
400701 RAM
400702 RAM
400703 RAM
400704 RAM
400705 RAM
400706 RAM
400707 RAM
400708 RAM
400709 RAM
400710 RAM
400711 RAM
400712 RAM
400713 RAM
400714 RAM
400715 RAM
400716 RAM
400717 RAM
400718 RAM
400719 RAM
400720 RAM
400721 RAM
400722 RAM
400723 RAM
400724 RAM
400725 RAM
400726 RAM
400727 RAM
400728 RAM
400729 RAM
400730 RAM
400731 RAM
400732 RAM
400733 RAM
400734 RAM
400735 RAM
400736 RAM
400737 RAM
400738 RAM
400739 RAM
400740 RAM
400741 RAM
400742 RAM
400743 RAM
400744 RAM
400745 RAM
400746 RAM
400747 RAM
400748 RAM
400749 RAM
400750 RAM
400751 RAM
400752 RAM
400753 RAM
400754 RAM
400755 RAM
400756 RAM
400757 RAM
400758 RAM
400759 RAM
400760 RAM
400761 RAM
400762 RAM
400763 RAM
400764 RAM
400765 RAM
400766 RAM
400767 RAM
400768 RAM
400769 RAM
400770 RAM
400771 RAM
400772 RAM
400773 RAM
400774 RAM
400775 RAM
400776 RAM
400777 RAM
400778 RAM
400779 RAM
400780 RAM
400781 RAM
400782 RAM
400783 RAM
400784 RAM
400785 RAM
400786 RAM
400787 RAM
400788 RAM
400789 RAM
400790 RAM
400791 RAM
400792 RAM
400793 RAM
400794 RAM
400795 RAM
400796 RAM
400797 RAM
400798 RAM
400799 RAM
400800 RAM
400801 RAM
400802 RAM
400803 RAM
400804 RAM
400805 RAM
400806 RAM
400807 RAM
400808 RAM
400809 RAM
400810 RAM
400811 RAM
400812 RAM
400813 RAM
400814 RAM
400815 RAM
400816 RAM
400817 RAM
400818 RAM
400819 RAM
400820 RAM
400821 RAM
400822 RAM
400823 RAM
400824 RAM
400825 RAM
400826 RAM
400827 RAM
400828 RAM
400829 RAM
400830 RAM
400831 RAM
400832 RAM
400833 RAM
400834 RAM
400835 RAM
400836 RAM
400837 RAM
400838 RAM
400839 RAM
400840 RAM
400841 RAM
400842 RAM
400843 RAM
400844 RAM
400845 RAM
400846 RAM
400847 RAM
400848 RAM
400849 RAM
400850 RAM
400851 RAM
400852 RAM
400853 RAM
400854 RAM
400855 RAM
400856 RAM
400857 RAM
400858 RAM
400859 RAM
400860 RAM
400861 RAM
400862 RAM
400863 RAM
400864 RAM
400865 RAM
400866 RAM
400867 RAM
400868 RAM
400869 RAM
400870 RAM
400871 RAM
400872 RAM
400873 RAM
400874 RAM
400875 RAM
400876 RAM
400877 RAM
400878 RAM
400879 RAM
400880 RAM
400881 RAM
400882 RAM
400883 RAM
400884 RAM
400885 RAM
400886 RAM
400887 RAM
400888 RAM
400889 RAM
400890 RAM
400891 RAM
400892 RAM
400893 RAM
400894 RAM
400895 RAM
400896 RAM
400897 RAM
400898 RAM
400899 RAM
400900 RAM
400901 RAM
400902 RAM
400903 RAM
400904 RAM
400905 RAM
400906 RAM
400907 RAM
400908 RAM
400909 RAM
400910 RAM
400911 RAM
400912 RAM
400913 RAM
400914 RAM
400915 RAM
400916 RAM
400917 RAM
400918 RAM
400919 RAM
400920 RAM
400921 RAM
400922 RAM
400923 RAM
400924 RAM
400925 RAM
400926 RAM
400927 RAM
400928 RAM
400929 RAM
400930 RAM
400931 RAM
400932 RAM
400933 RAM
400934 RAM
400935 RAM
400936 RAM
400937 RAM
400938 RAM
400939 RAM
400940 RAM
400941 RAM
400942 RAM
400943 RAM
400944 RAM
400945 RAM
400946 RAM
400947 RAM
400948 RAM
400949 RAM
400950 RAM
400951 RAM
400952 RAM
400953 RAM
400954 RAM
400955 RAM
400956 RAM
400957 RAM
400958 RAM
400959 RAM
400960 RAM
400961 RAM
400962 RAM
400963 RAM
400964 RAM
400965 RAM
400966 RAM
400967 RAM
400968 RAM
400969 RAM
400970 RAM
400971 RAM
400972 RAM
400973 RAM
400974 RAM
400975 RAM
400976 RAM
400977 RAM
400978 RAM
400979 RAM
400980 RAM
400981 RAM
400982 RAM
400983 RAM
400984 RAM
400985 RAM
400986 RAM
400987 RAM
400988 RAM
400989 RAM
400990 RAM
400991 RAM
400992 RAM
400993 RAM
400994 RAM
400995 RAM
400996 RAM
400997 RAM
400998 RAM
400999 RAM
401000 RAM
```

LIST

GROUP

@84

P4

[illegible]

```
POKE 23728,Y-coordinate
POKE 23729,X-coordinate
RANDOMIZE USR FN a()
```

You will notice that a manual or automatic attempt at LISTing the line 1 REM statement causes a spurious error report. This occurs because the MC code contains values that the system uses for color code markers. This causes annoying interference with program development and editing. Luckily we can circumvent the problem by doing the following...

1. LIST 2
2. POKE 2368, 30
3. Avoid the following...
 - a. LIST
 - b. LIST 1
 - c. moving the edit cursor onto line 1

FIG. 1

```
058: 176: 092: 047: 214: 080: 245: 230:
063: 245: 203: 063: 203: 063: 203: 063:
033: 224: 063: 071: 004: 017: 032: 000:
025: 016: 253: 241: 230: 007: 071: 014:
000: 009: 241: 001: 000: 000: 254: 128:
048: 006: 254: 064: 048: 003: 024: 002:
009: 009: 229: 033: 177: 092: 078: 006:
003: 062: 000: 203: 057: 203: 031: 016:
250: 000: 005: 203: 063: 015: 252: 230:
007: 006: 000: 225: 009: 071: 004: 014:
127: 203: 001: 016: 252: 126: 161: 119:
201:
```

Sum=8422

LONG ISLAND GROUP
Sinclair
TIMEX

If you got the correct value delete program lines 10 through 80 and SAVE the rest of the program. With the two lines of this program MERGED into any other you UNPLOT using the following procedure....

SOFTWARE/HARDWARE RECEIVED

COMPASS

Two 2068 programs on one cassette, one is an assembler, the other a compiler. The compiler was reviewed in Syntax July/84. I basically agree with that review (a nice try, but too limited). If you have HOT Z or Zeus and would like to write a comparison review of these assemblers, let us know and we'll provide you with this submission. These are from JRC (John Richard Coffey), POB 448, Scottsbury, Indiana 471070. (812)-752-6071

WINKY BOARD 2000

The Winky Board helps you load/save and copy tapes, by "conditioning" the signals to and from your computer and tape recorder. The 2000 seems to do the job, and while it won't solve all your problems, I feel no timeowner should be without one. G. Russel Electronics, RD 1 Box 539, Centre Hall, Pa. 16828

EVEREST ASCENT

A strategy type word game, Everest Ascent pits your wits against Mount Everest. You must choose a team of sherpas and the right equipment and supplies necessary to reach the top. Originally not runnable, because it was a Spectrum program (all in BASIC), we have adjusted it to run on the 2058. Richard Shephard Software (UK).

VENDORS

Sunset Electronics
2254 Taraval Street
San Francisco, Ca. 94116
(415) 665-8330

One of the most complete catalogs. They have almost everything for your ZX/TS & 2068. Price is reasonable.

Zebra Systems
78-06 Jamaica Avenue
Woodhaven, N.Y. 11421
(212) 296-2385
(212) 296-2229 BBS

Zebra's line is just about as comprehensive as Sunset. Note the free Bulletin Board Service (BBS)

Secondbase
700 Lexington Avenue
Altoona, Pa. 16601
(814) 942-1871

Keyboard overlay for the TS/1000. Gives you raised keys - \$15.00.

WMJ Data Systems
4 Butterfly Drive
Hauppauge, N.Y. 11788

Household & Business software for the ZX & 2068

Microaide
1401 Oak Tree Road
Iselin, N.J. 08830
(201) 283-1910

Out of Warranty repairs
(Henry Garcia)

Quick Silva Inc.
426 West Nakoma
San Antonio, Texas 78216
(512) 340-3684

Games, Games & more Games

WHAT IS LIST

LIST is a not-for-profit User's Group, dedicated to the exchange of information and ideas, about Timex/Sinclair computers, the expansion of computer literacy and public service to anyone interested in TS computers.

L.I.S.T. is published once/month and covers the actions of the group, programs, reviews and articles of interest to ZX/TS computer users. Membership in LIST is \$12.00/year and includes subscription to L.I.S.T.ing. Membership year is February to January. Sample copy sent upon receipt of large SASE.

Copies provided on exchange basis with other bona fide user groups.

Free copies available for pickup at 10 Idle Day Drive, while supplies last.

COPYRIGHT NOTICE

All articles published in this periodical are the property of LIST (Long Island Sinclair Timex) Group. No portion of any of these articles may be reproduced in any form without the express written consent of LIST.

LATE BREAKING NEWS

- 1) Finally got in touch with Al Levy of LICA. He will come to our September meeting and give us a presentation on LICA.
- 2) COMPUTERS & ELECTRONICS - Magazine reports that Sinclair is negotiating with Samsung Electronics, South Korea, for the manufacture of ZX81's and Spectrums.

LIST IS AT:

Box 438
Centerport, N.Y. 11721

QUIRKS & QUANDARIES

Using Jeff Street's "EXPND" program turned up an interesting foible in our 2068. You may recall that the graphic character "I" on the Spectrum prints out as the keyword "STICK" in some BASIC programs (e.g., Transylvanian Tower).

However, when we put "STICK" in a REM statement for EXPND to operate upon, we get the original graphic line character. There must be different tables operating here.

LIST GROUP

PE